COLLABORATIVE DESIGN OF MODELS

Peter Rittgen
Vlerick Leuven Gent Management School and University of Borås
Reep 1, 9000 Gent, Belgium and Allégatan 1, 50190 Borås, Sweden

ABSTRACT

We have used a design science approach to study collaborative modeling of information systems. Following this we have first designed a collaborative modeling architecture based on the empirical findings in a modeling study using conventional tools. A specific tool for this architecture was then developed and used as a vehicle to collect data for the consolidation of the architecture and for further theorizing on information systems design as a communicative practice.

KEYWORDS

Collaborative Modeling, Unified Modeling Language, Model Negotiation, Participatory Modeling.

1. INTRODUCTION

Design science is often viewed as complementary to and sometimes even competing with the more "traditional" behavioral science approach to information systems research. In their seminal paper on design science (Hevner, March, Park, & Ram, 2004), the authors argue that the aim of behavioral science is to build and test theories of social behavior whereas design science "seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts" (ibid., p. 75). But design science might also be seen as an extension of behavioral science. The latter, when used in information systems research, typically also deals with artifacts, even if indirectly, by studying IT-mediated organizational action. But contrary to design science the artifact itself is not touched by the researcher. The behavioral scientist is primarily interested in the effects that the artifact has on social behavior.

Design science expands this scenario by allowing the researcher to manipulate the artifact. It includes possibilities that range from the creation of completely new artifacts to adapting, changing, extending and integrating existing ones. As a consequence the researcher has a considerably larger impact on shaping organizational development. But design science as introduced by (March & Smith, 1995) is also an extension of the more restricted view on design where the artifact is the aim rather than the instrument of research. According to this restricted view a prototype is a result in itself that visualizes research ideas and is not necessarily subjected to empirical investigation. The existence of it alone is sufficient to demonstrate the feasibility of a suggested approach and to justify the validity of claims regarding its usefulness.

The broader view on design science embeds the development of the artifact into circles of empirical and theoretical validation and improvement. This ensures both rigor and relevance of the results and puts design science on an equal footing with the more accepted behavioral paradigm. Figure 1 shows the principle layout of design science research.

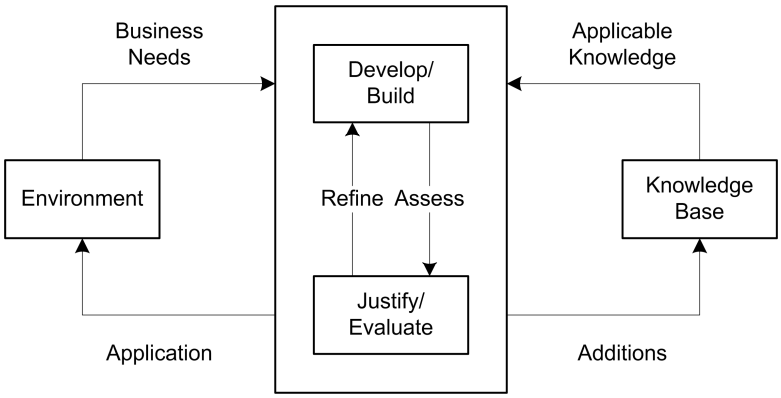

Figure 1. The design science circles (adapted from Hevner et al., MISQ 28:1)

At the core is the design circle where an artifact is built. Artifacts can be constructs, models, methods or instantiations. The artifact is evaluated in an empirical study and the results will lead to a refinement (i.e., further development) of the artifact, and so on. The design is driven by business needs, and the subsequent application of the artifact in an appropriate environment (e.g., a real-life organization) provides empirical validation of the relevance of the artifact together with further or changed needs that are considered in the next round (relevance circle). The final circle (rigor circle) embeds the artifact design into a knowledge creation process. Existing theoretical knowledge is applied in the design of the artifact and its evaluation updates or adds knowledge to the base.

In the following we describe the use of this approach to collaborative modeling as a communicative practice. We performed two iterations of the inner circle and one iteration each of the outer circles. The first iteration of the inner circle is about building and evaluating a framework or architecture for collaborative modeling (a model artifact according to the design science terminology). This activity is detailed in section 2, COMA – COllaborative Modeling Architecture. The design of this architecture artifact makes use of theoretical knowledge from the field of organizational semiotics. This constitutes the upper half of the right circle (applicable knowledge).

The second iteration of the inner circle builds on the results of the first. From the COMA architecture, which is a model artifact in terms of design science, we have derived an instantiation. Instantiations in design science are concrete exemplars, e.g. instances of models or prototypes. In our case it is a tool that embodies (or implements) the architecture. The COMA tool, as an instantiation artifact, allows us to put the abstract artifact of the architecture to a concrete use by subjecting it to empirical evaluation (lower half of the left circle, application). This evaluation will then enable us to draw conclusions on the architecture as well. The COMA tool is in this sense also a vehicle for validating the underlying architecture, which accounts for the lower half of the right circle (additions to the knowledge base).

Section 3, The COMA Tool, elaborates on the development of the tool. Section 4, Experiences from Deploying the COMA Tool, covers the lessons we learned from the practical application of the tool and the theoretical insights we gained from this.

2. COMA – COLLABOARATIVE MODELING ARCHITECTURE

Collaborative modeling processes have been studied by (Bommel, Hoppenbrouwers, Proper, & Weide, 2006; Frederiks & Weide, 2006; Hoppenbrouwers, Lindeman, & Proper, 2006; Hoppenbrouwers, Proper, & Weide, 2005; Persson, 2001). In this research modeling has been characterized as a largely communicative practice. We have continued in this tradition and have chosen a theoretical background that is closely associated with communicative practice. For the development of the COllaborative Modeling Architecture (COMA) we have therefore made use of two major sources: the theoretical knowledge from organizational semiotics as, for example, embodied in the semiotic ladder (Stamper, 1991) and the empirical results of controlled modeling experiments with students.

We conducted 3 experiments that involved a total of 26 groups of 2-3 students in informatics over a period of 3 years. The students were provided with a textual description of four business processes in a

hospital. They were asked to model these processes with the help of two different modeling languages that they could choose freely from a set of four languages: ARIS-EPC (Scheer, 1999), FMC-Petri nets (Keller & Wendt, 2003), UML (OMG, 2004, 2006), and DEMO (Dietz, 1999). Based on the results of these experiments we derived a layered meta-model of the modeling process that includes a model of the negotiation process.

To understand the modeling process, we assumed that two factors are predominant in model creation:

- The internal mental processes of each modeler, and
- The conversations between modelers and within the group.

To get access to the former we used a think-aloud process-tracing methodology (Ericsson & Simon, 1993; Srinivasan & Te´eni, 1995) where the observants speak out what they are currently thinking. The utterances were then transcribed yielding the think-aloud protocols. The same is done with the conversations. In addition to that we also considered the product of the modeling process, the models themselves, to fill the gaps in the protocols and to help with interpreting ambiguous phrases in them. Open issues that could not be dealt with in this way were marked on the coding scheme and clarified by ex-post interviews with the respective groups.

To develop a preliminary coarse-grain categorization we used the upper four 'rungs' of the semiotic ladder: syntactic, semantic, pragmatic, and social. They refer to the structure of sign systems (e.g., a language), the meaning of the signs, their use, and the norms of a community, respectively. An initial coding phase within this framework revealed that the syntactic and semantic levels, which together make up the language level, are divided into the natural language domain and the modeling language domain depending on the kind of language used to describe the business.

The activities on the pragmatic level were classified as 'Understanding' and 'Organizing the Modeling Process'. The former term was then further refined into 'Understanding the language' and 'Understanding the text'; the latter can be divided into 'Setting the agenda' and 'Negotiation'. The social level consists of rules for acceptance and rejection in the negotiation. A detailed discussion of these categories can be found in (Rittgen, 2007). Here we present only the results concerning the social and pragmatic levels. The coarse-grained overall structure is summarized in Figure 2.

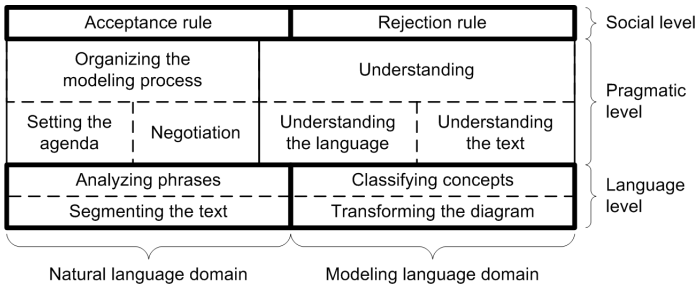

Figure 2. Levels and domains of collaborative modeling

2.1. Results on the Social Level

The social norms within a modeling team are mainly made up of rules for determining whether a proposal is accepted or rejected. We observed that these rules do not have to be logical complements which allows for situations where a proposal can be neither rejected nor accepted but requires further convincing to decide one way or the other. A termination rule was applied occasionally to force a decision if a negotiation got stuck, i.e., when there were no more changes in the individuals' convictions over an extended period of time. We witnessed two types of rules:

- Rules of majority, where a certain number of group members had to support or oppose a proposal in order for the whole group to accept or reject it (e.g., more than half). A tie-break rule was sometimes specified (e.g., for the case of an equal number of supporters and opponents). The tie-break could involve seniority issues.
- Rules of seniority, where the weight of a group member's support or opposition was related to his or her status within the group. This status could be acquired (e.g., by experience) or

associated with a position to which the member was appointed. A frequent example of this was the case of a more experienced modeler who was considered as the leader by the group and took decisions on their behalf. The other members filled the role of consultants in such a case.

These rules were sometimes set up explicitly before the group began their work, or in an early phase of this work. But in most cases they rather emerged as the result of each member's behavior. Individuals making regular contributions of high quality were likely to acquire seniority. In homogeneous teams majority rules were used more often.

2.2. Results on the Pragmatic Level

On the pragmatic level we discovered two distinct types of behavior, each of which can be classified in two sub-categories (the abbreviations of the categories are used as indices of the respective coded terms later on):

- Understanding, which concerns the text of the case description (index UT) or the (modeling) language (index UL), and
- Organizing the modeling process, which involves two types of activities: setting the agenda (index SA) and negotiation (index N)?

Understanding was established by questions and answers. If the respondent could not provide clarification, an assumption was made. Agendas have been used by the participants in our study as an instrument for roughly structuring the modeling session. They were introduced in the beginning and then adapted during the session if necessary. On the whole most groups started by reading the case description completely and then organized their work around the flow of the text.

The majority of the activities on the pragmatic level were associated with negotiation, though. This is surprising as modeling is typically rather pictured as an intuitive act that is largely the product of a creative brain (e.g., a consultant) that possibly receives some input from other stakeholders in the modeling process (e.g., domain experts from the respective departments).

According to our results modeling is a relatively well-structured process. It consists of a limited number of well-defined activities on all levels of the semiotic ladder. We are aware that further research will reveal more activities but from the experience of the three experiments that yielded a decreasing number of new ones, we are confident that the total number of activities will converge. The activities identified so far can therefore be assumed to be relatively stable. To a certain extent this is even true across different modeling languages, although the terminology of concepts may vary and not every concept is realized in each of the languages.

An analysis of the workflows on the pragmatic level revealed a structure that goes beyond the mere identification of generic activities. We found out that the negotiation process actually follows a certain pattern. This pattern is shown in Figure 3.

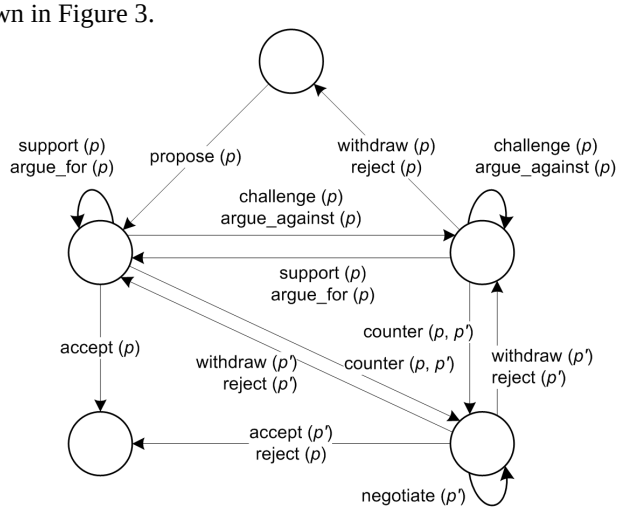

Figure 3. Negotiation pattern

It consists of an initial and reject state at the top, a state where acceptance is favored (upper left-hand corner), a state where rejection is favored (upper right-hand corner), a recursive sub-state for negotiating a counter-proposal (lower right-hand corner) and an accept state (lower left-hand corner). Each of the states allows for a set of certain pragmatic activities that take the negotiation to a different state. We have left out the parameters concerning the modeler who performs the activity and the argument (if present). In general any modeler can perform any activity but there are a few rules to be observed. A modeler making a proposal is implicitly assumed to support it. He is the only one who may withdraw it. A counter-argument is brought up by a different modeler but a counter-proposal can also be made by the proponent of the original proposal, e.g., to accommodate counter-arguments. With the help of the pattern of Figure 3 we can control the negotiation component of a modeling support system. On the other levels we were not able to discover an equally strong pattern of activities. This will affect the kind of support a tool can provide at the language level.

3. THE COMA TOOL

Our analyses of the modeling sessions showed us that modeling is a complex process involving issues such as collective sense-making, negotiations and group decisions. It is therefore worthwhile to consider tool support for this process. This is particularly true in an interorganizational setting where participants are often geographically distributed. The tool we envision helps group members in understanding the modeling situation, creating and discussing modeling alternatives, and deciding on the best one, all in a shared internet-based environment. The following paragraphs elaborate on the components that such a tool should provide.

The architecture of a collaborative modeling tool, i.e., a system that supports a group in developing models, is still under investigation. Some authors have suggested groupware systems that help teams in collective sense-making (Boehm, Grunbacher, & Briggs, 2001; Briggs, de Vreede, & Nunamaker, 2003; Conklin, Selvin, Buckingham Shum, & Sierhuis, 2003; Hoppenbrouwers, Lindeman, & Proper, 2006) which is an important part of the modeling process. (Conklin, Selvin, Buckingham Shum, & Sierhuis, 2003) reports on an approach, Compendium, that is the result of 15 years of experience. Compendium combines three different areas: meeting facilitation, graphical hypertext and conceptual frameworks. To make them work, facilitation is viewed as essential to remove the cognitive overhead for the group members, i.e., the necessity to develop hypertext literacy, which cannot be assumed in all participants.

On the technology side, the critical elements are question-based templates, metadata and maps. They allow participants to move freely between different levels of abstraction and formalization as the need dictates. The question-based templates guide the process by supplying relevant questions, the answers to which will lead the group towards a better understanding of the problem and towards the development of appropriate solutions (e.g., models). The metadata is used to provide additional information that is also considered relevant but was not anticipated in the templates or lies at the intersection of templates. The maps have a hierarchical structure and the same concept can appear in different maps so that its use in different contexts can be understood. This feature is called transclusion.

Groupware systems for collective sense-making, as the one mentioned, address an important issue in collaborative modeling. They can therefore be used as the core of a collaborative modeling tool. So far these systems are typically tailored for specific modeling languages though (in the case of Compendium, World Modeling Framework and Issue-Based Information System). For a collaborative modeling tool they need to be more modular so that any modeling language can be “plugged in” (e.g., other enterprise or information systems modeling languages). In addition, there is also the need for a negotiation component that facilitates structured arguments and decisions regarding modeling choices. The model shown in Figure 3 can function as a workflow template controlling such a negotiation component.

The COMA architecture assumes the existence of a modeling tool that takes care of the syntactic level of the modeling process (i.e., a conventional single-user tool). Such tools exist in abundance for most modeling languages. Some of the tools even provide support on the semantic layer, e.g. by determining run-time properties of a business process such as liveness and freedom of deadlocks.

COMA is therefore primarily situated on the pragmatic and social levels. On the former it specifies distributed model negotiation, which can be seen as a special case of a distributed decision support system (Aiken, Vanjani, & Krosp, 1995). This is described in section 3.1.

On the social level it provides facilities to express the social norms that the group has adopted and that control the way in which negotiation takes place, in particular under which conditions a proposed change to the model is accepted. This is described in section 3.2.

3.1. Negotiation in the COMA tool

Distributed model negotiation means the coordination of the efforts of a number of modellers. The results suggest that such a system must provide the following functions:

- Propose, Withdraw, Counter
- Support, Argue_for
- Challenge, Argue_against

To simplify the architecture we have decided to arrange the negotiation activities in three groups of related functions concerning proposals, supports and challenges. This implies that we drop the distinction of only two competing proposals (proposal and counter-proposal) in favor of considering any number of competing proposal (one from each proponent / modeler). It also means that we view withdrawals as less important as they can be replaced by making a new proposal (if necessary, an empty one). We have also dropped the distinction between Support and Argue_for because the former is the same as the latter without supplying the argument. The same holds for Challenge and Argue_against. The challenger is required to provide an argument (i.e., an indication of how the proposal can be improved), the supporter not.

In addition to the functions the negotiation component needs to provide each modeler with a clear overview of: The model building process and the current status of the negotiation. A view of the model building process involves:

- The current stable version of the model as agreed upon so far
- A version that contains the local changes made by the respective group member (the local model editor), and
- The proposals made by the other modelers suggesting changes to the current version.

Regarding the status of the negotiation process the modeler needs access to the following information:

- What are the arguments for and against a proposal?
- Who is in support of or against a proposal?
- What is the final decision regarding the proposal?

The model building view is divided into three areas. The upper one shows the current version that has emerged from the negotiation process so far. It is used as a reference for all other temporary versions such as the proposals and the local version. This means that suggested changes are always expressed in relation to the current version.

The lower left area contains the local version, i.e. it serves as the model editor for the group member running this particular copy of the modeling support system. It offers the model creation facilities pertaining to the syntactic level such as introducing and connecting nodes. In addition to that it provides the pragmatic functions related to making proposals. Making a proposal implies that the local version is published, i.e. made accessible to the other modelers.

The negotiation view shows a list of pros and cons for the proposal (called supports and challenges). These are the arguments supplied when the argue_for or argue_against function is performed on a proposal by a modeler. Depending on the currently active rule on the social level this determines overall acceptance or rejection of the proposal, e.g. when a required majority has been reached. A facilitator might also force a decision if the required condition cannot be reached.

An accepted proposal makes all competing proposals obsolete so they will be deleted. Observe that the deleted proposals still exist as local copies so that the respective modelers can decide to post them again as proposals if applicable. If a proposal is accepted it becomes the new current version, i.e. the upper area is updated for all group members.

The architecture that we have described so far is a direct consequence of the results from the empirical study detailed in the previous sections. The architecture supports the activities on the social, pragmatic and syntactic levels. We have built a prototype that implements this architecture and that allows us to gain further insight into the modeling process. This prototype can be employed in a number of different ways:

- It can be used to test the suggested architecture and thereby indirectly confirm the study results in a broader empirical study.

- The additional tool support in this study makes it easier to perform such a study on a larger scale, e.g. with distributed team members, and
- It also provides additional information about the modeling process that was not available in the original study.

The existence of version histories, for example, makes it possible to analyze the modeling process in a more detailed manner regarding the development stages of a model. Another example is the negotiation log that gives us a deep insight into the arguing process and the competition between different model alternatives. A study supported by this tool can therefore also contribute to the development of new theories of the modeling process.

On the practical side, the prototype can also help in detecting shortcomings and suggesting improvements. These suggestions can be related to the implementation (i.e., the tool itself) or the architecture behind it (as outlined above). Issues such as the design of the user interface and migration to other modeling languages are important considerations here. Section 8 describes the COMA tool.

3.2 Social Norms in the COMA tool

COMA distinguishes between two rules to decide on the acceptance of proposals: Rules of majority and rules of seniority. When a rule of majority is applied, the modeling team operates in a democratic (unfacilitated) mode where each modeler has a vote of usually the same weight (different weights might be considered, though). Whether a proposal is accepted only depends on how many supporters and challengers it has. The rule specifies the minimum number of supports required, and the maximum number of challenges allowed for a proposal to be accepted. Prior to the modeling session the team has to agree on suitable parameters. The required number of supports should be at least two to avoid that a single modeler (e.g. the proponent) can make the decision alone. A maximum number of challenges of 0 would force a unanimous decision.

When a rule of seniority is applied, the modeling team has a facilitator that makes the decision. Other group members cannot directly influence the decision, but they can do so indirectly by making suitable comments (i.e., supports and challenges). The facilitator can (and indeed should) consider the supports and challenges in his or her decision. The facilitator is either determined externally or by the group (e.g., the most experienced modeler). In principle two or more facilitators might be appointed for larger groups but they have to coordinate their decisions in order to avoid confusion.

4. DEPLOYING THE COMA TOOL

We have deployed the COMA tool in a case study that involved collaborative modeling in UML, primarily use case and activity diagrams with the ultimate purpose of requirements elicitation. The study was performed during the summer 2007 in a large company manufacturing mobile network components (called MobCom henceforth). The overall project aimed at the introduction of a new warehouse management system to replace the existing scattered landscape of “island solutions”. The objective of the first phase of that project was to derive the requirements for such a system, and as part of that work an analysis of the business processes of the logistics unit (customer distribution centre). As MobCom works with the Unified Modeling Language (UML) internally, the analysis models were supposed to be in UML, too. We therefore decided to model the logistics processes with the help of activity diagrams and to develop use cases for the requirements specification.

As the overall business process is quite complex (the Arrival of Goods alone subsumes more than 150 activities), the modeling work was split up into manageable chunks, each of which was handled by a group of two to three modelers depending on chunk size. The goal of each group member was not to work on a separate part of the group’s chunk but to collectively work on the whole model.

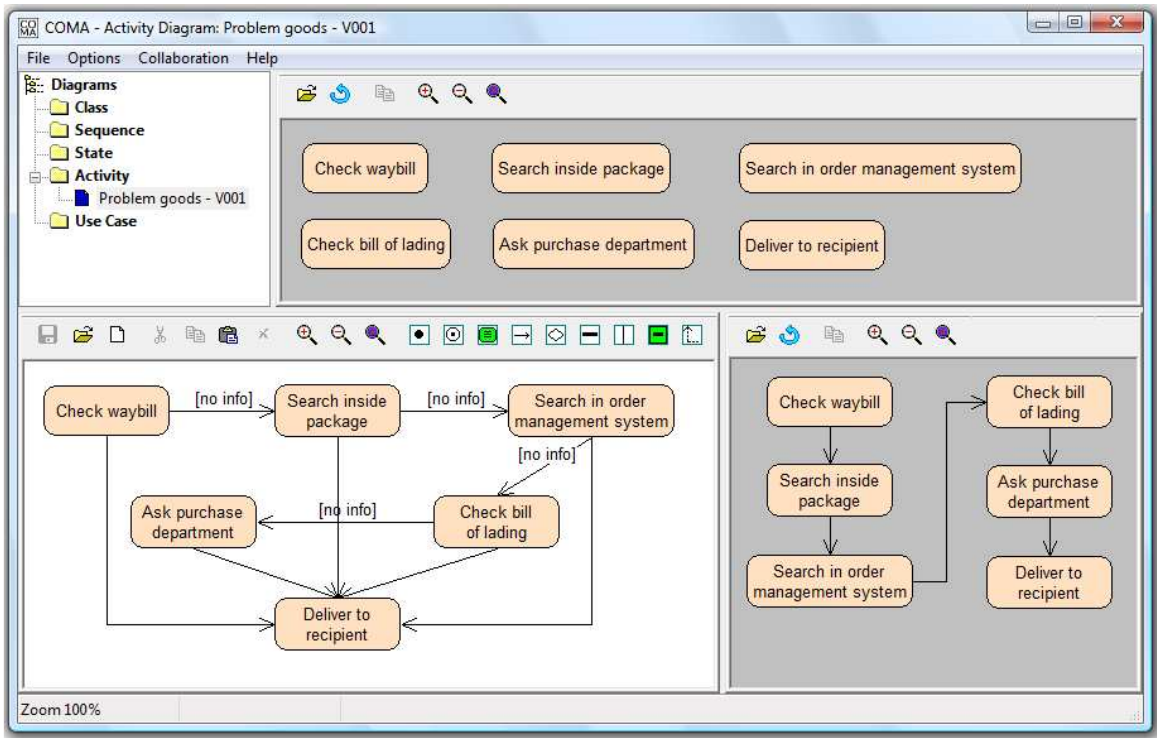

Figure 4. COMA screenshot

Figure 4 shows a snapshot of the modeling process at a certain stage. This is supposed to give the reader an example of how modeling in COMA proceeds and how it helps with a particular problem, namely that of Resolving a conflict. The group in question was concerned with the handling of so-called problem goods, i.e. goods with an unclear recipient. In a first step they simply wrote down all the activities that are involved thus arriving at the first version (upper pane). One member suggested to order the activities in a certain sequence and made a respective proposal (lower right pane). He knew from experience that this was indeed the order in which the activities were carried out at the Arrival unit. Another modeler agrees with the principle sequence but he is quite sure that the search for the recipient is terminated as soon as the recipient is identified and further steps are skipped. He draws the respective diagram in his editor window (lower left pane) and makes a counter-proposal. On seeing the apparent conflict the first modeler confirms with the operations staff that this is indeed the case and withdraws his original proposal in favor of the new one. The new proposal was subsequently adopted by the group as version two.

To find problems in modeling in a systematic way we conducted interviews with all participants. The focus was on open questions regarding the types of problems that were encountered in collaborative modeling. This was complemented by closed questions on the support that the COMA tool provided in solving these problems. A simple three-point Likert scale was used (good/some/no support). Table 1 summarizes the results. The support column shows the mode of the sample regarding the support variable.

Table 1. Results from the case study

Modeling issue	**Support by COMA tool**
Resolving a conflict, i.e. clarifying a misunderstanding	Good support
Making sense, i.e. trying to understand a situation within or without the modeling language	Some support
Conceptualizing a situation, i.e. expressing a situation in the modeling language	Good support
Communicating a view, i.e. making an individual view accessible to others	Good support
Aligning views, i.e. making different proposals converge	Good support

Clarifying an issue, i.e. getting help with an unclear issue from others	Some support
Discussing a problem, i.e. trying to structure an unstructured problem and arriving at a potential solution alternative	No support
Managing a modeling project, i.e. assigning tasks to groups and monitoring the results	No support
Evaluating alternatives, i.e. assessing the relative merits of each proposed solution	Good support
Agreeing on a solution, i.e. arriving at a common version of a model	Good support
Ensuring progress, i.e. making sure that we proceed towards a result, e.g. a complete model	Some support

The first issue, resolving a conflict, has already been discussed in the example. The participants in the case study found that the tool provided good support here. Making sense is less supported because it involves issues that are outside the modeling sphere and that require the group members to resort to natural language. Good support is provided for conceptualizing a situation where representation of a situation in terms of the modeling language is achieved with the help of the model editor. The same can be said for communicating a view and aligning views, which are directly supported by respective functionalities of the tool, namely the proposal and negotiation functions. Participants were also in need of help from others when clarifying an issue but found only some support for this problem by looking at the solutions proposed by others. They got no help with issues such as discussing a problem and managing a modeling project. Evaluating alternatives and agreeing on a solution, however, were suitably supported by the negotiation component according to the case study participants. As another important issue they named ensuring progress. Here COMA allowed for the appointment of a group leader who could force decisions and speed up the process.

In summary we can say that the COMA architecture and tool have proved useful in practice. The practitioners stated that the tool did indeed provide substantial support that goes significantly beyond that of conventional modeling tools and that the functions of the COMA tool address important concerns they had when modeling in a group. But we also found new business needs that are not addressed by the current architecture. Some issues in collaborative modeling require the use of natural language and project management issues are also relevant. This concludes an iteration of the relevance circle of design science (left circle in Figure 1).

On the theoretical level the study has confirmed the validity of the existing components of the COMA architecture, i.e. the modeling, proposal and negotiation components. But it has also shown the need for further components in the architecture. We need at least two more: a natural language and a project management component. Preliminary results suggest that the former could be provided by integrating conventional groupware functionality such as email, chat, brainstorming and the like. The theoretical results so far constitute an iteration of the rigour circle (right circle in Figure 1) but further iterations are necessary. This includes a larger study to confirm the results on a broader basis.

5. CONCLUSIONS

We followed a design-science approach to study collaborative modeling as a communicative practice. In the design circle we developed and improved two artifacts, an architecture called COMA and a tool that implements this architecture. The design was inspired by theoretical insights from organizational semiotics and driven by observations of group modeling behaviour. In the relevance circle we put the tool to a practical test to assess the usefulness of the tool and the validity of the architecture and to find further business needs that can fuel the next round of the design circle. Here we primarily identified the need for natural language and project management support. In the rigour circle we confirmed the knowledge about the existing architecture components (modeling, proposal and negotiation) and found two new ones by evaluating the experiences with the tool. According to this it seems vital to add groupware functionality to the tool as many problems of the modellers can only be solved with the help of (structured) conversations. Project management was also mentioned as an important issue that requires further research.

REFERENCES

AIKEN, M., VANJANI, M. & KROSP, J. (1995) Group decision support systems. Review of Business, 16, 38-42.

BOEHM, B., GRUNBACHER, P. & BRIGGS., R. O. (2001) Developing Groupware for Requirements Negotiation: Lessons Learned. IEEE Software, 18, 46-55.

BOMMEL, P. V., HOPPENBROUWERS, S. J. B. A., PROPER, H. A. E. & WEIDE, T. P. V. D. (2006) Exploring Modeling Strategies in a Meta-modeling Context. IN MEERSMAN, R., TARI, Z. & HERRERO, P. (Eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France. Berlin, Germany, Springer.

BRIGGS, R. O., DE VREEDE, G. J. & NUNAMAKER, J. (2003) Collaboration Engineering with Thinklets to Pursue Sustained Success with Group Support Systems. Journal of MIS, 19, 31-63.

CONKLIN, J., SELVIN, A., BUCKINGHAM SHUM, S. & SIERHUIS, M. (2003) Facilitated Hypertext for Collective Sensemaking: 15 Years on from gIBIS. IN WEIGAND, H., GOLDKUHL, G. & DE MOOR, A. (Eds.) Proceedings of the 8th International Working Conference on the Language-Action Perspective on Communication Modeling (LAP'03). Tilburg, The Netherlands.

DIETZ, J. L. G. (1999) Understanding and modeling business processes with DEMO. IN AKOKA, J., BOUZEGHOUB, M., COMYN-WATTIAU, I. & MÉTAIS, E. (Eds.) Proceedings of the 18th International Conference on Conceptual Modeling ER '99. Berlin, Springer.

ERICSSON, K. & SIMON, H. (1993) Protocol Analysis: Verbal Reports as Data, Boston, MIT Press.

FREDERIKS, P. J. M. & WEIDE, T. P. V. D. (2006) Information Modeling: the process and the required competencies of its participants. Data & Knowledge Engineering, 58, 4-20.

HEVNER, A. R., MARCH, S. T., PARK, J. & RAM, S. (2004) Design Science in Information Systems Research. MIS Quarterly, 28, 75-105.

HOPPENBROUWERS, S. J. B. A., LINDEMAN, L. & PROPER, H. A. (2006) Capturing Modeling Processes - Towards the MoDial Modeling Laboratory. IN MEERSMAN, R., TARI, Z. & HERRERO, P. (Eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France. Berlin, Germany, Springer.

HOPPENBROUWERS, S. J. B. A., PROPER, H. A. & WEIDE, T. P. V. D. (2005) Formal Modeling as a Grounded Conversation. IN GOLDKUHL, G., LIND, M. & HARALDSON, S. (Eds.) Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modeling (LAP`05), Kiruna, Sweden. Linköping and Borås, Linköpings Universitet and Högskolan i Borås.

KELLER, F. & WENDT, S. (2003) FMC: An Approach Towards Architecture-Centric System Development. 10th IEEE Symposium and Workshop on Engineering of Computer Based Systems. Pasadena, CA, IEEE Computer Society.

MARCH, S. T. & SMITH, G. (1995) Design and Natural Science Research on Information Technology. Decision Support Systems, 15, 251-266.

OMG (2004) UML 2.0 Superstructure Specification. Needham; MA, OMG.

OMG (2006) Unified Modeling Language: Infrastructure. Needham, MA, OMG.

PERSSON, A. (2001) Enterprise Modeling in Practice: Situational Factors and their Influence on Adopting a Participative Approach. Department of Computer and Systems Sciences, Stockholm University.

RITTGEN, P. (2007) Negotiating Models. IN KROGSTIE, J., OPDAHL, A. & SINDRE, G. (Eds.) Advanced Information Systems Engineering, 19th International Conference, CAiSE 2007, Trondheim, Norway, June 2007, Proceedings. Berlin, Springer.

SCHEER, A.-W. (1999) ARIS - Business Process Modeling, Berlin, Springer.

SRINIVASAN, A. & TE´ENI, D. (1995) Modeling as Constrained Problem Solving: An Empirical Study of the Data Modeling Process. Management Science, 41, 419-434.

STAMPER, R. (1991) The Semiotic Framework for Information Systems Research. IN NISSEN, H., KLEIN, H. & HIRSCHHEIM, R. (Eds.) Information Systems Research: Contemporary Approaches and Emergent Traditions. Amsterdam, North-Holland.